

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-259115

(43)Date of publication of application : 13.09.2002

(51)Int.Cl.

G06F 5/06
G06F 12/02
G06F 13/38

(21)Application number : 2001-384346

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 18.12.2001

(72)Inventor : YASOJIMA HIROYUKI

(30)Priority

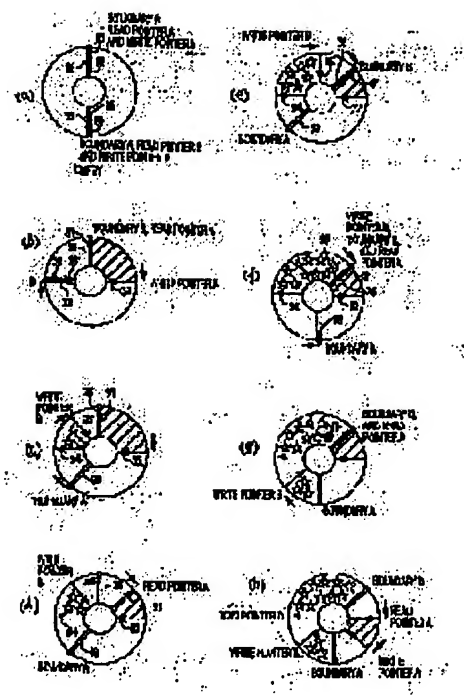
Priority number : 2000 739265 Priority date : 19.12.2000 Priority country : US

(54) FIFO MEMORY

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an FIFO memory device capable of minimizing an unused memory in device operation.

SOLUTION: The FIFO memory device is provided with a memory array having a plurality of address arrangements and capable of storing input data and a boundary pointer indicating the end point of a buffer area to be formed in the memory array capable of storing the input data and characterized by controlling the value of the boundary pointer in accordance with the volume of the input data. In the constitution, a memory space can be utilized at maximum especially by controlling the value of the boundary pointer in accordance with the volume of the input data.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-259115

(P2002-259115A)

(43) 公開日 平成14年9月13日 (2002.9.13)

(51) Int. Cl. ⁷	識別記号	F I	テラコード (参考)
G 0 6 F 5/06		G 0 6 F 5/06	B 5 B 0 6 0 C 5 B 0 7 7
12/02	5 8 0	12/02	5 8 0 D
13/38	3 1 0	13/38	3 1 0 C 3 1 0 D

審査請求 未請求 請求項の数20 O L (全 8 頁)

(21) 出願番号 特願2001-384348 (P2001-384348)

(22) 出願日 平成13年12月18日 (2001. 12. 18)

(31) 優先権主張番号 09/739265

(32) 優先日 平成12年12月19日 (2000. 12. 19)

(33) 優先権主張国 米国 (U S)

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 八十島 広至

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(74) 代理人 100097445

弁理士 岩崎 文雄 (外2名)

Fターム (参考) 5B060 AB20

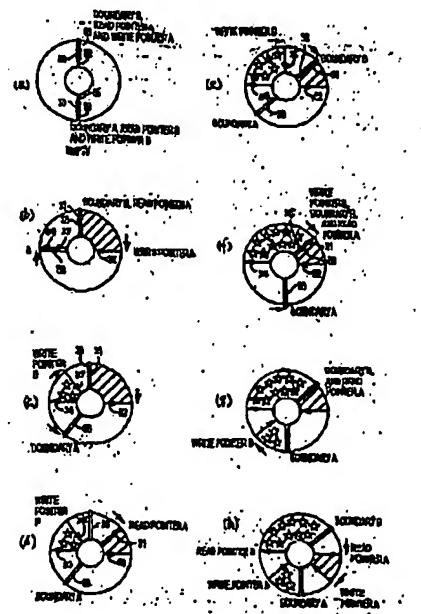
5B077 DD04 DD15

(54) 発明の名称 F I F Oメモリ

(57) 【要約】

【課題】 本発明はデバイス動作において未利用メモリを最小限化するF I F Oメモリデバイスを提供することを目的とする。

【解決手段】 複数のアドレス配置を有し、入力データを格納するメモリアレイと、前記入力データが格納される前記メモリアレイ内に形成されるバッファ領域の終了点を示すバウンダリポインタと、格納されている前記入力データの量に応じて前記バウンダリポインタの値を調整することを特徴とする。この構成のうち、特に入力データの量に応じてバウンダリポインタの値を調整することにより、メモリスペースを最大限に利用することが可能となる。



【特許請求の範囲】

【請求項1】 複数のアドレス配置を有し、入力データを格納するメモリアレイと、前記入力データが格納される前記メモリアレイ内に形成されるバッファ領域の終了点を示すバウンダリポインタと、格納されている前記入力データの量に応じて前記バウンダリポインタの値を調整するコントローラとを備えるFIFOメモリ。

【請求項2】 メモリアレイに接続され、バッファ領域の読出しアドレスを示す読出しポインタと、前記メモリアレイに接続され、前記バッファ領域の書き込みアドレスを示す書き込みポインタとを備える請求項1記載のFIFOメモリ。

【請求項3】 メモリアレイがリングバッファを利用して実現されることを特徴とする請求項1記載のFIFOメモリ。

【請求項4】 入力データの量に1:1対応することを基本にバッファサイズを増加するために、コントローラがバウンダリポインタを動かせることを特徴とする請求項3記載のFIFOメモリ。

【請求項5】 格納されている入力データの量に応じて、コントローラがバウンダリポインタの値を動的に変えることを特徴とする請求項1記載のFIFOメモリ。

【請求項6】 複数のアドレス配置を有し、入力データを格納するメモリアレイと、前記入力データが格納される前記メモリアレイ内に形成される第1のバッファ領域の終了点を示す第1のバウンダリポインタと、前記入力データが格納される前記メモリアレイ内に形成される第2のバッファ領域の終了点を示す第2のバウンダリポインタと、格納されるべき前記入力データの量に応じて前記第1および第2のバウンダリポインタの値を調整するコントローラとを備えるFIFOメモリ。

【請求項7】 メモリアレイに接続され、第1のバッファ領域の読出しアドレスを示す第1の読出しポインタと、前記メモリアレイに接続され、前記第1のバッファ領域の書き込みアドレスを示す第1の書き込みポインタと、前記メモリアレイに接続され、第2のバッファ領域の読出しアドレスを示す第2の読出しポインタと、前記メモリアレイに接続され、前記第2のバッファ領域の書き込みアドレスを示す第2の書き込みポインタとを備える請求項6記載のFIFOメモリ。

【請求項8】 メモリアレイがリングバッファを利用して実現されることを特徴とする請求項6記載のFIFOメモリ。

【請求項9】 入力データの量に1:1対応することを基本に第1および第2のバッファサイズを増加するために、コントローラが第1および第2のバウンダリポインタを動かせることを特徴とする請求項8記載のFIFOメモリ。

【請求項10】 格納されている入力データの量に応じて、コントローラが第1および第2のバウンダリポイン

タの値を動的に変えることを特徴とする請求項6記載のFIFOメモリ。

【請求項11】 複数のアドレス配置を有し、入力データを格納するメモリアレイを配するステップ、前記入力データが格納される前記メモリアレイ内に形成されるバッファ領域の終了点を示すバウンダリポインタを定義するステップ、格納されている前記入力データの量に応じて前記バウンダリポインタの値を調整するステップを備えるFIFOメモリのデータ格納方法。

【請求項12】 バッファ領域の読出しアドレスを示す読出しポインタを定義するステップ、前記バッファ領域の書き込みアドレスを示す書き込みポインタを定義するステップを備えることを特徴とする請求項11記載のFIFOメモリのデータ格納方法。

【請求項13】 メモリアレイがリングバッファを利用して実現されることを特徴とする請求項11記載のFIFOメモリのデータ格納方法。

【請求項14】 入力データの量に1:1対応することを基本にバッファサイズを増加するために、バウンダリポインタが変更されることを特徴とする請求項13記載のFIFOメモリのデータ格納方法。

【請求項15】 格納されている入力データの量に応じて、バウンダリポインタの値を動的に変えられることを特徴とする請求項11記載のFIFOメモリのデータ格納方法。

【請求項16】 複数のアドレス配置を有し、入力データを格納するメモリアレイを配するステップ、前記入力データが格納される前記メモリアレイ内に形成される第1のバッファ領域の終了点を示す第1のバウンダリポインタを定義するステップ、前記入力データが格納される前記メモリアレイ内に形成される第2のバッファ領域の終了点を示す第2のバウンダリポインタを定義するステップ、格納されるべき入力データの量に応じて前記第1及び第2のバウンダリポインタの値を調整するステップを備えるFIFOメモリのデータ格納方法。

【請求項17】 第1のバッファ領域の読出しアドレスを示す第1の読出しポインタを定義するステップ、前記第1のバッファ領域の書き込みアドレスを示す第1の書き込みポインタを定義するステップ、第2のバッファ領域の読出しアドレスを示す第2の読出しポインタを定義するステップ、前記第2のバッファ領域の書き込みアドレスを示す第2の書き込みポインタを定義するステップを備えることを特徴とする請求項16記載のFIFOメモリのデータ格納方法。

【請求項18】 メモリアレイがリングバッファを利用して実現されることを特徴とする請求項16記載のFIFOメモリのデータ格納方法。

【請求項19】 入力データの量に1:1対応することを基本にバッファサイズを増加するために、第1及び第2のバウンダリポインタが調整されることを特徴とする

請求項18記載のFIFOメモリのデータ格納方法。

【請求項20】 格納されている入力データの値に応じて、コントローラが第1および第2のバウンダリポイントの値を動的に変えることを特徴とする請求項16記載のFIFOメモリのデータ格納方法。

【発明の詳細な説明】

【発明の属する技術分野】本発明は、ファースト・イン・ファースト・アウト（First-in, First-out (FIFO)）メモリに関するもので、特に、システム動作中に異なるセクタ（バッファ）のサイズ（深さ）の変化を許容する動的プログラム可能なFIFOメモリに関するものである。

【従来の技術】FIFOメモリバッファは従来からよく知られており、一般的には各々が独立したクロック信号の制御により動作する2つの別のデータシステム間におけるデータ転送に利用される。多くのものは、データを送る側のデータシステムがFIFOメモリに書き込み、一方、データを受ける側のデータシステムは書き込まれた順序通りにFIFOメモリからデータを読み出す仕組みとなっている。また、リングバッファ構造を利用してFIFOを構成することも知られている。リングバッファの動作に従えば、メモリの固定領域は読み出しポイント（RP）及び書き込みポイント（WP）を利用することにより継続的、循環的にアクセスされ得る。図1は一般的なリングバッファメモリの動作を示す概念図である。図1（a）は空状態のバッファを示している。図において、バッファが「空状態」のとき、読み出しポイント2と書き込みポイント4は同じ位置に存在する。しかし、図1（b）に示すように、バッファにデータが書き込まれると、書き込みポイント4は時計回りに進む。書き込みポイント4のアドレスは書き込み動作が終了するときに1つずつ進むことにより、書き込みポイント4は次に利用可能なメモリ位置にセットされる。なお、データがリングバッファに読み込まれるとき、読み出しポイント2は動かない。図1（c）はデータがバッファに書き込まれ、かつバッファから読み出されるときの例を示す概念図である。図において、データがバッファに書き込まれるとき、書き込みポイント4は進み続ける。一方、データがバッファから読み出されるとき、読み出しポイント2は進む。読み出しポイント2のアドレスは読み出し動作が完了すると1つずつ進むことにより、読み出しポイント2は読み出される次のメモリ位置にセットされる。図1（d）はリングバッファが「満杯」の状態の例を示す概念図である。図において、データは利用可能なメモリの全てに書き込まれている。ここで、バッファから読み出されることを必要とされるデータが上書きされないように、コントローラ（図示せず）はさらなる書き込みポイント4の増加、およびさらなるデータ書き込みを防ぐ。メモリ利用を最大にするのと同様、別個のデータ区切りを処理するアプリケーションを実現するために、シングル・リン

グバッファ内における多数のFIFO動作の配置も知られている。例えば、それぞれが最大1kbyte必要とするデータA及びデータBを格納しようとしたとき、データA及びデータBがメモリ1kbyte以下であれば、単一の1kbyteのFIFOを利用することが可能となる。図2（a）は多数のデータ区切りを格納することが可能なFIFOメモリの例を示した概念図である。図において、FIFOは2つの書き込みポイントWPa、WPb、2つの読み出しポイントRPa、RPbを備え、リングバッファ中に2つのFIFO（すなわち、FIFO A、FIFO B）を構成する。各読み出し/書き込みポイント（例：RPa/WPa）は図1に示した読み出しポイント2および書き込みポイント4と同じように、動作する。図2（a）のリングバッファは、データA、データB両方の格納を許容し、図2（a）（b）の例に示すように、FIFO A（RPa/WPa）、FIFO B（RPb/WPb）は共にデータを含む。上記構成によれば、たとえデータAとデータBが最大1kbyteの格納容量を必要とした場合でも、通常動作時はデータAおよびデータBの実際的に必要な格納容量は1kbyte以下なので、データA及びデータBは共に利用可能なメモリスペースを1kbyteしか持たない図2（a）に示されるリングバッファに格納され得る。

【発明が解決しようする課題】多数の読み出し・書き込みポイントを持つ上記FIFOが配分されるメモリスペースの全体的な効率と利用を高めようとするにもかかわらず、様々な不足と問題が依然として存在する。特に、このようなデバイスでは、各読み出し/書き込みポイントは一般的に、与えられたデータ区切りが書き込まれ得るFIFO内においてスペースの量を定義又は制限する機能を備えるバウンダリと連携している。例えば、バウンダリは各データ区切りがFIFOメモリの1/2分余裕を持つようにセットされる。一旦、バウンダリがセットされると、デバイスの動作中は固定され、調整不可能となる。このような固定されたバウンダリを使用すると、しばしば利用可能なメモリスペースを十分に利用できない。例えば、データAがFIFO Aを完全に満たしたとき、たとえFIFO Bが空状態であっても、データAがFIFO Bに格納され得ないため、データ区切りAについてのデータ格納は停止する。図2（b）は2つのFIFOリングバッファの例を示す概念図である。図において、読み出しポイントBはFIFO Aのバウンダリとして機能し、読み出しポイントAはFIFO Bのバウンダリとして機能する。言い換えれば、書き込みポイントAが読み出しポイントBに届く前に、データAはFIFO Aに書き込まれ得る。もし、読み出しポイントBが進めば、追加データはFIFO Aに書き込まれることができる。FIFO Bも同様に、FIFO Bのバウンダリとして動作する読み出しポイントAとともに動作す

る。図2(b)に示されるFIFOの動作もまた利用可能なメモリスペースを十分に利用できない。例えば、図2(b)を見てみると、追加データがFIFO Aに書き込まれ、FIFO Aの書き込みポインタはFIFO Bの読み出しポインタに到達するとき、システムコントローラは追加データがFIFO Aに書き込まれないように動作する。図2(b)に示されるように、FIFO Bは追加データを格納可能な未利用メモリスペースをまだ有している。しかし、残念ながらこの利用可能なメモリスペースはそのデバイスのバウンダリ構造では、FIFO Aに配置することができない。他の従来技術としては、米国特許第5,097,442に記載されているような、特殊なアプリケーションに従ってFIFOに与える深さを予めプログラムしておく例がある。このデバイスでは、プログラマーに予め定義された部分のみをFIFOとしての利用に割り当てることができるようにしておき、残りの部分は他の目的の利用に割り当てることにより、全体的なメモリ構造をより効率的に使用することを可能としている。この米国特許では、深さポインタが読み出しポインタに予め決められた固定値を加えたものに等しくなるように設定されている。上記の既知のFIFOデバイスは使用を指定されたメモリスペースを巧みに取り扱いを可能としているが、依然としてFIFOメモリの効率的な利用をさらに最大化することが必要とされている。上記説明のように、前記各FIFOデバイスは動作シナリオを有し、FIFOデバイスに割り当てられたメモリスペースの相当な部分は未利用のままである。したがって、FIFOデバイスの動作においてこの未利用メモリを最小限化することによりデバイスの全体的な効率的利用を高めるFIFOデバイスとFIFOメモリコントローラが求められている。上記課題を解決すべく、本発明はデバイス動作において未利用メモリを最小限化するFIFOメモリデバイスを提供することを目的とする。

【課題を解決するための手段】上記目的を達成するために、本発明のFIFOメモリは、複数のアドレス配置を有し、入力データを格納するメモリアレイと、前記入力データが格納され得る前記メモリアレイ内に形成されるバッファ領域の終了点を示すバウンダリポインタと、格納されている前記入力データの量に応じて前記バウンダリポインタの値を調整することを特徴とする。この構成のうち、特に入力データの量に応じてバウンダリポインタの値を調整することにより、メモリスペースを最大限に利用することが可能となる。さらに、入力データの量に1:1対応することを基本にバッファサイズを増加するために、コントローラがバウンダリポインタを動かせることにより、メモリスペースの利用を最大化することが可能となる。

【発明の実施の形態】以下、本発明のFIFOにおける実施形態について図面を参照しながら説明する。図3は

本発明のFIFOメモリデバイスの全体構成図である。図において、FIFOメモリ10は、データ(すなわち、書き込みデータ)を受け入れ、データ(すなわち、読み出しデータ)を出力する手段を有するリングバッファ12で構成される。リングバッファ12は、FIFO A、FIFO Bと呼ぶ2つの独立したFIFOメモリで構成される。デバイス10は複数のポインタ14から構成され、それらの機能は各FIFOの読み出しポインタおよび書き込みポインタを定義するのはもちろん、各FIFOのバウンダリを定義するものである。すなわち、FIFO Aに関するポインタは(1)FIFO Aから読み出すFIFO A内の次のメモリ位置を示す読み出しポインタA(RPa)20、(2)入力データを受け入れるFIFO A内の次のメモリ位置を示す書き込みポインタA(WPa)22、FIFO Aに割り当てられたメモリスペースの終点を示すバウンダリポインタ(Ba)24である。同様に、FIFO Bに関するポインタは(1)FIFO Bから読み出すFIFO B内の次のメモリ位置を示す読み出しポインタB(RPb)26、(2)入力データを受け入れるFIFO B内の次のメモリ位置を示す書き込みポインタB(WPb)28、FIFO Bに割り当てられたメモリスペースの終点を示すバウンダリポインタ(Bb)30である。なお、それらポインタを形成するために、例えばレジスタや他の適当なメモリデバイスを利用することができる。デバイス10はさらに、コントローラ13を備える。コントローラ13はFIFOポインタ14に接続され、各FIFOを最大限利用すべくFIFO AおよびFIFO Bのサイズを動的に定義する機能を有するものであり、マイクロプロセッサや他の適当なプロセッサユニットを利用して形成することができる。各FIFOに関する読み出しポインタおよび書き込みポインタの読み出しや更新に関する動作は従来のFIFOメモリと同様の方法で実行されるので、ここでは特に説明を要しない。上述したように、バウンダリAで定義されるメモリ位置は、FIFO Aに割り当てられるメモリスペースの終点を定義するものである。同様に、バウンダリBで定義されるメモリ位置は、FIFO Bに割り当てられるメモリスペースの終点を定義するものである。本発明によれば、FIFO AまたはFIFO Bのいずれかへの入力データ(すなわち、データAまたはデータB)の量に連動して、コントローラ13はバウンダリA(すなわち、バウンダリポインタBa)およびバウンダリB(すなわち、バウンダリポインタBb)の値を動的に変更する。入力データの量に基づいてFIFO AおよびFIFO Bのサイズを動的に変更することによって、リングバッファ12に含まれるメモリスペースの利用を向上させることができる。さらに、データが所定のFIFOに読み込まれると、コントローラ13は受け入れたデータの量を決定して、予め決められた量だけバッ

フェ領域（すなわち、FIFO AまたはFIFO B）を拡大しようとする。例えば、データがFIFO Aに読み込まれた場合、コントローラ13はバウンダリAの値をリセットすることによりFIFO Aを拡大する。バウンダリAがFIFO Bに格納されたデータと隣接しない（バウンダリAを進めるためにFIFO B内に利用可能スペースが存在するような）場合、バウンダリAの全体的なメモリスペースに余裕を持たせるようにリセットされる。しかし、FIFO Bが“満杯”であれば、コントローラ13はバウンダリAをリセットしようとはしない。FIFOの状態（空状態、満杯、半分満たされている状態）は、対象のFIFOの読み出し／書き込みポイントの値と現在のバウンダリの値を調べることによりコントローラ13が容易に判断できる。もちろん、上述したシナリオとは異なる方法も可能である。例えば、FIFO Aに入力された前のデータAの影響でバウンダリAが進んでいて、FIFO Aからデータが読み出されていないため、バウンダリBが現在進めないようなときに、FIFO Aが“満杯”でなく、データBがFIFO Bに入力されれば、コントローラ13はバウンダリAを縮めるようにバウンダリAをリセットする。このことにより、FIFO Aに割り当てられたスペースを減少し、FIFO Bに割り当てられるスペースを増加する。バウンダリAを縮める量は、FIFO Bに入力されるデータの量により決定される。上記説明の通り、コントローラ13が所定のバウンダリを伸ばしたり縮めたりする量はFIFOに入力されるデータ量に応じて決定される。例えば、図4（a）～図4（h）で明確にされている本発明の実施形態では、1:1対応している。しかし、他のやり方をすることも可能である。例えば、入力されるデータ量とは関連のない量だけバウンダリを動かすようにコントローラ13をプログラムすることも可能である。また、入力されるデータ量に応じて1:1以外の対応でバウンダリを動かすようにコントローラ13をプログラムすることも可能である。さらに、FIFO AおよびFIFO Bのサイズが予め定義した最小サイズ以下に減少しないようにコントローラ13をプログラムすることも可能である。FIFO AおよびFIFO Bのバウンダリを動的に変更することにより、リングバッファ12に含まれる全体としてのメモリスペースを最大限利用することができる。実際に、利用可能なFIFOメモリの利用を最大限化するデータ処理を実現化するものである。特に、データAおよびデータBを生成するタスクが同時に実行されることが続かない場合であって、タスクAが実行されるとき、データAはおそらくFIFOメモリの大半のスペースを必要とする。本発明では、一定時間、FIFO Aの使用に優先的にメモリを割り当てて、それとは別の時間にFIFO Bの使用にメモリを割り当てる。このように、メモリ使用の効率を向上することにより、システムの実際のメ

モリスペースを増加させることなく、FIFO AとFIFO Bの利用可能メモリスペースを効果的に増加させることができる。図4（a）～図4（g）は本発明のFIFOメモリデバイスの動作例を示している。上述のように、本発明のFIFOメモリデバイスは、FIFOメモリデバイスに割り当てられるメモリスペースを最大限利用できるように、入力データの量に応じて動的に移動可能なバウンダリを提供する。図4（a）は本発明を適用した代表的なリングバッファを示す図である。このリングバッファは2つのFIFOを有し、読み出しポイントA（RPa）31、書き込みポイントA（WPa）32、バウンダリポイントA33、読み出しポイントB（RPb）34、書き込みポイントB（WPb）35およびバウンダリポイントB36で構成される。図4（a）に示されるように、FIFO AまたはFIFO Bのいずれにもデータが存在しないとき、各FIFOは利用可能なメモリスペースが半分ずつ割り当てられる。RPa31とWPa32は同じ位置で、バウンダリポイントB（すなわち、FIFO BにデータBを格納可能な最新のメモリ位置）はRPa31で定義されている。ここで、FIFO Bで利用可能な最新のメモリ位置はRPa-1（RPaマイナス1）と同一のメモリアドレス位置である。RPb、WPbおよびバウンダリAは上記と同様に定義される。図4（b）からわかるように、データがFIFO Aに入力され始めると、WPa32は従来技術で説明したものと同じ方法で移動し始める。しかし、本発明では、RPb34、WPb35およびバウンダリA33も、FIFO Aに追加メモリスペースが割り当てられるように移動する。RPb34、WPb35およびバウンダリAの移動量はWPa32が入力データに用立てた量と同一である。結果として、図4（b）に示されるように、FIFO Aに格納されたデータは全体のメモリスペースの1/4を占めるので、RPb34、WPb35およびバウンダリA33も全体のメモリスペースの1/4分移動されている。FIFO Aに入力されるデータに応じてRPb34、WPb35およびバウンダリA33の移動量を1:1対応以外になるようにデバイスをプログラムすることも可能である。図4

（c）を参照しながら説明を続けると、データがFIFO Bに格納され始めると、WPb35の動きが示すように、システムはWPb35が移動した量とちょうど同じ量だけバウンダリA33を後方へ移動させようとする。バウンダリA33が後方へ移動することにより、FIFO Bの利用可能なメモリスペースが増加する。FIFO AとFIFO Bは共に“ラップ・アラウンド（wrap-around）”であると言える。言い換えれば、図4（c）に示されるように、FIFO Bにデータが格納され続け、FIFO Aから読み出されるデータがなく、データAがバウンダリBに隣接する（バウンダリBが移動できない）場合、WPb35がバウンダリB36

に到達すると、FIFO B内に利用可能なメモリスベ
ースがなお存在するので、データBはバウンダリA33
とRpb34の間に定義されるメモリ位置のFIFO
Bに格納され続けることとなる。図4(d)を参照しな
がら説明を続ける。RPa31の動きにより示されるよ
うに、データがFIFO Aから読み出された場合、バ
ウンダリA33またはバウンダリB36のいずれかの調
整はなくなる。しかし、図4(e)に見られるように、
新しいデータがFIFO Bに読み込まれた場合、RP
a31はバウンダリB36と隣接しないので、バウン
ダリB36は図4(b)においてバウンダリA33が移動
されたのと同じ方法で、入力データBの量と同じ量分移
動される。本実施例では、FIFO Bに入力されたデ
ータの量がバウンダリB36を移動可能なスペースを超
えるような場合、バウンダリB36は単純に最大量移動
される(バウンダリB36がRPa31に隣接するま
で)。また、新しく入力されたデータBに応じたスベ
ースと同じだけ、バウンダリB36の前方移動及びバウン
ダリA33の後方移動がなされるように、バウンダリA
33を後方へ移動させることも可能である(バウンダリ
A33がFIFO Aに格納されたいかなるデータとも
隣接していない場合)。続いて、図4(f)に示される
ように、Wpb35がバウンダリB36に隣接するよう
にFIFO Bに追加データが入力され続け、バウン
ダリB36がRPaに隣接する場合、バウンダリA33は
FIFO Bの利用可能なメモリを増加させるように移
動される。データAがバウンダリA33に隣接しない場
合、バウンダリA33は後方へ移動され得るのみであ
る。図4(g)に示されるように、FIFO Bに追加
データが書きこまれた場合、そのデータは隣接するバ
ウンダリA33を開始点として格納され、Wpb35の動
きは図に示されるように時計回りに進む。最後に、図4
(h)を見られるように、RPa31およびWpa32
の動きからわかるように、FIFO Aにさらなるデー
タが書きこまれたつ、FIFO Aからデータが読み出さ
れる場合、データBはバウンダリA33およびバウン
ダリB36の両方に隣接するため、バウンダリは移動しな
い。つまり、データが所定のバウンダリに隣接する場
合、そのバウンダリは移動されることはない。もちろ
ん、図4(a)～図4(h)に示したのはほんの一例で
あり、本発明はこれら図面の内容に限定されるものでは
ない。例えば、一のバウンダリAのみを動的に変更さ
れ、バウンダリBは固定とすることももちろん可能であ
る。また、入力データに応じてバウンダリが移動する量
は1:1対応以外のものでも実現可能であることもいう
までもない。多数の他の実現可能性が存在することも明
らかである。追加的なバリエーションも可能である。例
えば、本発明の適用はリングバッファに限らない。さら

に、読み出しポインタ、書き込みポインタおよびバウン
ダリの数を増やし、必要な方法で追加のポインタおよびバ
ウンダリの適用をコントローラにプログラムすることに
より、FIFOを追加(3個以上)することも可能であ
る。上述したように、FIFOメモリを動的にプログラ
ムする本発明により、従来技術からは得られない優位性
を有する。最も必要なのは、本発明のFIFOメモリデ
バイスなら、FIFOデバイス動作中の未利用のメモリ
を最小限化できることであり、このことにより、デバイ
スの効率利用を向上させることができる。さらには、F
IFOデバイスの効率を向上させることにより、FIFO
を利用するデバイスについて必要なメモリを低減化で
き、このことにより、コストおよびスペースの節約を実
現することができる。さらに付け加えると、動的に移動
可能なFIFOバウンダリによって、従来技術を使用し
た場合に要求された、設計者が仕様に必要なFIFOメ
モリの量を見積ろうとする必要性がなくなることにな
る。本発明の実施形態では実施適用の一例を開示した
が、本発明の技術思想を逸脱しない限り、種々の実施の
形態での実現が可能である。それゆえ、上記実施の形態
は、実例であり限定的なものではないあらゆる観点で考
慮されるべきものであり、特許請求の範囲に記載した内
容およびその均等物から発生するあらゆる変更事項も本
発明に含まれるものである。

【発明の効果】本発明のFIFOメモリは、複数のアド
レス配置を有し、入力データを格納するメモリアレイ
と、前記入力データが格納され得る前記メモリアレイ内
に形成されるバッファ領域の終了点を示すバウンダリポ
インタと、格納されている前記入力データの量に応じて
前記バウンダリポインタの値を調整することを特徴と
し、特に入力データの量に応じてバウンダリポインタの
値を調整することにより、メモリスペースを最大限に利
用することが可能となる。

【図面の簡単な説明】

【図1】リングバッファを利用した従来のFIFOメモリ
の通常動作を示す概念図

【図2】2つのFIFOメモリを有するリングバッファ
から構成される従来のFIFOメモリの通常動作を示す
概念図

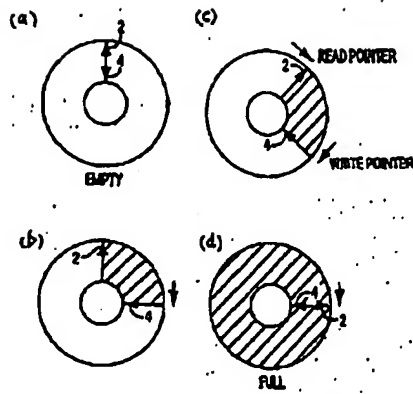
【図3】本発明のFIFOメモリデバイスの全体構成図

【図4】本発明のFIFOメモリの動作を示す図

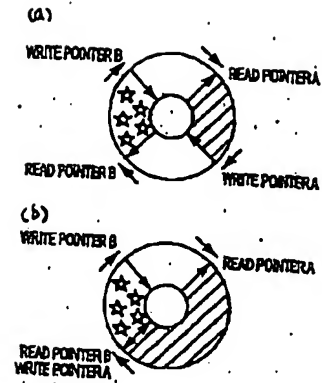
【符号の説明】

- 31 読み出しポインタA
- 32 書き込みポインタA
- 33 バウンダリポインタA
- 34 読み出しポインタB
- 35 書き込みポインタB
- 36 バウンダリポインタB

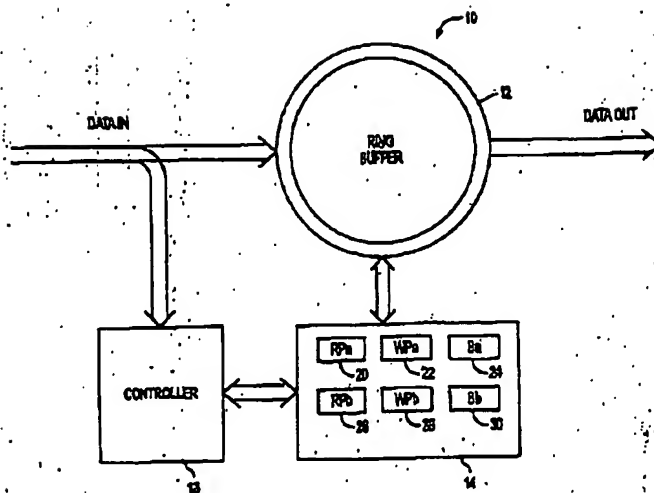
【図1】



【図2】



【図3】



【図4】

